Advanced Micro Devices

Advanced Media Framework – FRC

Programming Guide

Disclaimer

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information.

Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD, the AMD Arrow logo, ATI Radeon™, CrossFireX™, LiquidVR™, TrueAudio™ and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Windows™, Visual Studio and DirectX are trademark of Microsoft Corp.

Copyright Notice

Notice Regarding Standards. AMD does not provide a license or sublicense to any Intellectual Property Rights relating to any standards, including but not limited to any audio and/or video codec technologies such as MPEG-2, MPEG-4; AVC/H.264; HEVC/H.265; AAC decode/FFMPEG; AAC encode/FFMPEG; VC-1; and MP3 (collectively, the "Media Technologies"). For clarity, you will pay any royalties due for such third party technologies, which may include the Media Technologies that are owed as a result of AMD providing the Software to you.

MIT license

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

1 Introduction

AMF FRC is a technique for achieving high-end video frame rate conversion results from lower frame rate video inputs. This document provides a complete description of the AMD Advanced Media Framework (AMF) Video FRC Component. This component performs the frame generation.

2 AMF Video FRC Component

Video FRC accepts input frames stored in `AMFSurface` objects wrapping DirectX 11 or DirectX 12 textures. The output is placed in `AMFSurface` objects wrapping DirectX 11 or DirectX 12 textures, depending on the component configuration.

Include `public/include/components/FRC.h`

2.1 Component Initialization

The AMF Video FRC component should be initialized using the following sequence:

1. Create an AMF Context and initialize it for one of the following:
   i. DirectX 11
   ii. DirectX 12
2. Configure the FRC component by setting the necessary properties using the `AMFPropertyStorage::SetProperty` method on the FRC object.
3. Call the `AMFFRCImpl::Init` method of the video FRC object.

The details on component setup can be found from the document `AMF_API_Reference.pdf`

2.2 Configuring the FRC

The FRC supports the following input and output formats:

1. BRGA
2. NV12
3. RGBA
4. R10G10B10A2
5. RGBA_F16
6. P010

The output format must be same as the input and the format conversion is not supported. The parameters are set using the following properties:

Property Name	Type
AMF_FRC_ENGINE_TYPE	AMF_MEMORY_TYPE
AMF_FRC_MODE	AMF_FRC_MODE_TYPE

Property Name	Type
AMF_FRC_ENABLE_FALLBACK	Bool
AMF_FRC_INDICATOR	Bool
AMF_FRC_PROFILE	AMF_FRC_PROFILE_TYPE
AMF_FRC_MV_SEARCH_MODE	AMF_FRC_SEARCH_MODE_TYPE
AMF_FRC_USE_FUTURE_FRAME	Bool

Table 1. AMF FRC Properties

Name: `AMF_FRC_ENGINE_TYPE`

Values: `DX12` , `DX11`

Default Value: `DX12`

Description: Specifies the engine used to run shaders.

Name: `AMF_FRC_MODE`

Values:

Name	Description
`FRC_OFF`	Frame rate conversion is disabled. The frame data will be copied over to the output.
`FRC_ON`	Frame rate conversion is enabled. Note that the component will need to be called with a frame rate equal to double that of the input video.
`FRC_ONLY_INTERPOLATED`	Frame rate conversion is enabled. However, only the interpolated frames are returned.
`FRC_x2_PRESENT`	`AMF_REPEAT` will be returned for the QueryOutput() call with each source frame. The caller needs to call the FRC component with the same source frame again to get x2 frame rate.

Default Value: `FRC_ONLY_INTERPOLATED`

Description: Specifies which FRC frames are presented.

Name: `AMF_FRC_ENABLE_FALLBACK`

Values:

Name	Description
`true`	Low confidence to do the interpolation, two frames will be blended together.
`false`	Low confidence to do the interpolation, frame will be duplicated.

Default Value: `false`

Description: Specifies the fallback mode.

Name: `AMF_FRC_INDICATOR`

Values: `true` , `false`

Default Value: `true`

Description: Specifies whether or not the FRC indicator square is shown in the top right corner of the video.

Name: `AMF_FRC_PROFILE`

Values:

Name	Description
`FRC_PROFILE_LOW`	Less levels of hierarchical motion search. Only recommended for extremely low resolutions.
`FRC_PROFILE_HIGH`	Recommended for any resolution up to 1440p.
`FRC_PROFILE_SUPER`	More levels of hierarchical motion search. Recommended for resolutions 1440p or higher.

Default Value: `FRC_PROFILE_HIGH`

Description: Specifies the levels of hierarchical motion search. It is recommended to set this value according to the resolution of the input.

Name: `AMF_FRC_MV_SEARCH_MODE`

Values:

Name	Description
`FRC_MV_SEARCH_NATIVE`	Conduct motion search on the full resolution of source images.
`FRC_MV_SEARCH_PERFORMANCE`	Conduct motion search on the down scaled source images. Recommended for APU or low end GPU for better performance.

Default Value: `FRC_MV_SEARCH_NATIVE`

Description: Specifies the performance mode of the motion search.

Name: `AMF_FRC_USE_FUTURE_FRAME`

Values: `true` , `false`

Default Value: `false`

Description: When enabled, the information contained in the next frame in the sequence will be used in FRC interpolation calculations, in addition to the current pair of frames. This will introduce one extra frame time of latency.

2.3 Submitting Input and Retrieving Output

Once the FRC component is successfully initialized, you may start submitting input samples to it. Input samples must be submitted as `AMFSurface` objects.

At the same time poll for output by calling `AMFComponent::QueryOutput` on the FRC object. Polling for output samples can be done either from the same thread or from another thread.

Suspend submission of input samples briefly when `AMFComponent::SubmitInput` returns `AMF_INPUT_FULL`. Continue to poll for output samples and process them as they become available.

2.4 Terminating the FRC Component

To terminate the FRC component, call the `Terminate` method, or simply release the object. Ensure that the context used to create the FRC component still exists during termination.